|  |
| --- |
| B+ TREE IMPLEMENTATION |
|  |
|  |
| Abstract: The project to is understand the B+ tree indexing technique being implemented in the real world applications. Here the project is implemented using Minibase. |
|  |
| **Pruthvi & Vamsi** |
|  |

|  |  |
| --- | --- |
| **Contents:** |  |
| 1.Team Details | 3 |
| 2.Project Status  Insert() & \_insert():  Delete(): (Naïve Delete): | 3 |
| 3.File Description | 4 |
| 4.Logical Errors | 4 |
| 5.Division of work | 5 |
| 6.Output | 6 |

**PROJECT REPORT**

**1.Team Details:**

|  |  |
| --- | --- |
| **Name** | **UTA ID** |
| PRUTHVI RAJ MUTHYAM | 1001400715 |
| VAMSI KRISHNA YALAMANCHILI | 1001554490 |

**2.Project Status:**

The project is to implement the B+ tree. In this project we have to insert and delete the elements in the B+ tree. Initially, we understood how minibase works and referred Java Document to know the methods provided. As advised by the Professor and TA, we developed a proper algorithm for insertion and deletion of elements using insert(), \_insert() & delete() methods into the B+ tree. The description provided in the PDF helped us to formulate the algorithm properly. It took 1 week to build up the algorithm. After this, we started coding which too around 4-6 days to execute correctly.

We were able to implement all the three methods insert(), \_insert() and delete() methods successfully.

**Insert() & \_insert() :**

We created Insert function which takes <key, rid> as the parameter. Initially, the B+ tree created is empty and header page would be pointing to “Invalid Page”, if the header page is empty then we insert the value in leaf page and made header page point to newly created leaf page. If tree is not empty then we would be calling \_insert() method to check if there is space in the existing page and insert , else split the page accordingly. In \_insert() we will be checking the page is leaf or index before inserting the entries or splitting. We split the page into two if the index or leaf page is full. The existing page is divided into two and the second half is copied to a new page that we create and we update the pointers accordingly. As soon as the page is split the index is page update with <key, pageid>. We referred the PDF document to construct these methods.

**Delete() :**

We performed only Naïve delete. We looked into the PDF provided to get an idea of what to do. This involves the traversal of the tree based on key comparison to locate the key to be deleted and then simply removing key and data pointer. As we implemented only the Naïve delete, we didn’t have to worry about the balancing or redistribution of tree after the deletion operation. Coding for delete is not as complex as Insertion methods. As the insertion and deletion operations were working correctly, thus the implementation of the B+ tree is completed.

**3.File description:**

No new files were created during the project. The BTreeFile.java contains the source code for the insertion and deletion methods. BtTest.java file contains testing configurations to test the project.

**4. Logical Errors:**

* **Splitting the tree:** We faced many issues in the beginning as to how to split leaf/index pages. We tried some implementations but they didn’t work out well. After referring to the project description uploaded we were able to correctly handle the split of leaf/index pages.First we copied all the elements to a new page getting a record count while copying them and after by calculating a mid value copied half of the entries back to the existing leaf/index page.
* **Finding page to insert the given key value:** Initially we could not find a direct method to retrieve the page to insert the given key from the Minibase documentation. We tried to compare the given key value using BT.keyCompare() method with each and every page to find out the correct page for inserting the given key. We were not able to retrieve correct page ID’s. After TA discussed the method getPageNoBykey() in class which is not in the minibase documentation provided. We were able to directly find the page ID where we can insert the key which is returned by the method.
* **Checking whether a index/leaf page has enough space to insert a key:** We were unable to find whether there is enough space available in a index/leaf page to insert a given key as there is no direct method available for that in the minibase documentation provided. As advised by TA in the class we used getKeyDataLength() method which returns an integer value, comparing that with the index/leaf page available space we inserted the record accordingly.

**5. Division of work:**

Initially, we started the project by studying about insertion and deletion operation being performed in the B+ tree. Then we sat together and studied about the minibase and the java methods provided. We also contacted with the TA to get a clear idea regarding the implementation and clarify our doubts. After this, we formulated the algorithm step-wise using the functions provided. We checked regularly as we may not miss any of the minute detailsin implementing the methods. When we completed the algorithm using different methods required from the minibase, then we started coding. During coding we regularly debugged the algorithm to remove any errors. We divided the work as follows-

Pruthvi Raj Muthyam-

-Understood the Minibase and Java functions.

-Came up with the algorithm.

-Coding part.

-Commenting the code written as specified.

Yalamanchili Vamsi Krishna-

-Understood the Minibase and Java functions.

-Came up with the algorithm.

-Coding part.

-Reporting the project.

**6. Output:**

**Configuration and compilation on omega:**

Compiling BTreeFile.java

[pxm0715@omega ~]$ cd bplustree/

[pxm0715@omega bplustree]$ cd src

[pxm0715@omega src]$ cd btree

[pxm0715@omega btree]$ make

/opt/jdk1.6.0\_20/bin/javac-classpath/home/p/px/pxm0715/bplustree/lib/btreelib.jar:/home/p/px/pxm0715/bplustree:/home/p/px/pxm0715/bplustree/src \*.java

Compiling BTTest.java

[pxm0715@omega ~]$ cd bplustree/

[pxm0715@omega bplustree]$ cd src

[pxm0715@omega src]$ cd tests

[pxm0715@omega tests]$ make

/opt/jdk1.6.0\_20/bin/javac-classpath/home/p/px/pxm0715/bplustree/lib/btreelib.jar:/home/p/px/pxm0715/bplustree:/home/p/px/pxm0715/bplustree/src BTTest.java TestDriver.java

[pxm0715@omega tests]$ make bttest

/opt/jdk1.6.0\_20/bin/javac-classpath/home/p/px/pxm0715/bplustree/lib/btreelib.jar:/home/p/px/pxm0715/bplustree:/home/p/px/pxm0715/bplustree/src BTTest.java TestDriver.java

/opt/jdk1.6.0\_20/bin/java-classpath/home/p/px/pxm0715/bplustree/lib/btreelib.jar:/home/p/px/pxm0715/bplustree:/home/p/px/pxm0715/bplustree/src tests.BTTest

Output:

Replacer: Clock

Running tests....

IO error: java.io.IOException: Cannot run program "/bin/rm": CreateProcess error=2, The system cannot find the file specified

IO error: java.io.IOException: Cannot run program "/bin/rm": CreateProcess error=2, The system cannot find the file specified

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* The file name is: AAA0 \*\*\*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Print the B+ Tree Structure

[1] Print All Leaf Pages

[2] Choose a Page to Print

---Integer Key (for choices [3]-[5]) ---

[3] Insert a Record

[4] Delete a Record (Naive Delete)

[5] Delete some records (Naive Delete)

[6] Quit!

Hi, make your choice :3

Initially 100 values are inserted from 1 to 100.

-------------------------- MENU ------------------

[0] Print the B+ Tree Structure

[1] Print All Leaf Pages

[2] Choose a Page to Print

---Integer Key (for choices [3]-[5]) ---

[3] Insert a Record

[4] Delete a Record (Naive Delete)

[5] Delete some records (Naive Delete)

[6] Quit!

Hi, make your choice :0

---------------The B+ Tree Structure---------------

1 5

2 3

2 4

2 6

--------------- End ---------------

-------------------------- MENU ------------------

[0] Print the B+ Tree Structure

[1] Print All Leaf Pages

[2] Choose a Page to Print

---Integer Key (for choices [3]-[5]) ---

[3] Insert a Record

[4] Delete a Record (Naive Delete)

[5] Delete some records (Naive Delete)

[6] Quit!

Hi, make your choice :1

---------------The B+ Tree Leaf Pages---------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 3

Left Link : -1

Right Link : 4

0 (key, [pageNo, slotNo]): (1, [ 1 1 ] )

1 (key, [pageNo, slotNo]): (2, [ 2 2 ] )

2 (key, [pageNo, slotNo]): (3, [ 3 3 ] )

3 (key, [pageNo, slotNo]): (4, [ 4 4 ] )

4 (key, [pageNo, slotNo]): (5, [ 5 5 ] )

5 (key, [pageNo, slotNo]): (6, [ 6 6 ] )

6 (key, [pageNo, slotNo]): (7, [ 7 7 ] )

7 (key, [pageNo, slotNo]): (8, [ 8 8 ] )

8 (key, [pageNo, slotNo]): (9, [ 9 9 ] )

9 (key, [pageNo, slotNo]): (10, [ 10 10 ] )

10 (key, [pageNo, slotNo]): (11, [ 11 11 ] )

11 (key, [pageNo, slotNo]): (12, [ 12 12 ] )

12 (key, [pageNo, slotNo]): (13, [ 13 13 ] )

13 (key, [pageNo, slotNo]): (14, [ 14 14 ] )

14 (key, [pageNo, slotNo]): (15, [ 15 15 ] )

15 (key, [pageNo, slotNo]): (16, [ 16 16 ] )

16 (key, [pageNo, slotNo]): (17, [ 17 17 ] )

17 (key, [pageNo, slotNo]): (18, [ 18 18 ] )

18 (key, [pageNo, slotNo]): (19, [ 19 19 ] )

19 (key, [pageNo, slotNo]): (20, [ 20 20 ] )

20 (key, [pageNo, slotNo]): (21, [ 21 21 ] )

21 (key, [pageNo, slotNo]): (22, [ 22 22 ] )

22 (key, [pageNo, slotNo]): (23, [ 23 23 ] )

23 (key, [pageNo, slotNo]): (24, [ 24 24 ] )

24 (key, [pageNo, slotNo]): (25, [ 25 25 ] )

25 (key, [pageNo, slotNo]): (26, [ 26 26 ] )

26 (key, [pageNo, slotNo]): (27, [ 27 27 ] )

27 (key, [pageNo, slotNo]): (28, [ 28 28 ] )

28 (key, [pageNo, slotNo]): (29, [ 29 29 ] )

29 (key, [pageNo, slotNo]): (30, [ 30 30 ] )

30 (key, [pageNo, slotNo]): (31, [ 31 31 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 4

Left Link : 3

Right Link : 6

0 (key, [pageNo, slotNo]): (32, [ 32 32 ] )

1 (key, [pageNo, slotNo]): (33, [ 33 33 ] )

2 (key, [pageNo, slotNo]): (34, [ 34 34 ] )

3 (key, [pageNo, slotNo]): (35, [ 35 35 ] )

4 (key, [pageNo, slotNo]): (36, [ 36 36 ] )

5 (key, [pageNo, slotNo]): (37, [ 37 37 ] )

6 (key, [pageNo, slotNo]): (38, [ 38 38 ] )

7 (key, [pageNo, slotNo]): (39, [ 39 39 ] )

8 (key, [pageNo, slotNo]): (40, [ 40 40 ] )

9 (key, [pageNo, slotNo]): (41, [ 41 41 ] )

10 (key, [pageNo, slotNo]): (42, [ 42 42 ] )

11 (key, [pageNo, slotNo]): (43, [ 43 43 ] )

12 (key, [pageNo, slotNo]): (44, [ 44 44 ] )

13 (key, [pageNo, slotNo]): (45, [ 45 45 ] )

14 (key, [pageNo, slotNo]): (46, [ 46 46 ] )

15 (key, [pageNo, slotNo]): (47, [ 47 47 ] )

16 (key, [pageNo, slotNo]): (48, [ 48 48 ] )

17 (key, [pageNo, slotNo]): (49, [ 49 49 ] )

18 (key, [pageNo, slotNo]): (50, [ 50 50 ] )

19 (key, [pageNo, slotNo]): (51, [ 51 51 ] )

20 (key, [pageNo, slotNo]): (52, [ 52 52 ] )

21 (key, [pageNo, slotNo]): (53, [ 53 53 ] )

22 (key, [pageNo, slotNo]): (54, [ 54 54 ] )

23 (key, [pageNo, slotNo]): (55, [ 55 55 ] )

24 (key, [pageNo, slotNo]): (56, [ 56 56 ] )

25 (key, [pageNo, slotNo]): (57, [ 57 57 ] )

26 (key, [pageNo, slotNo]): (58, [ 58 58 ] )

27 (key, [pageNo, slotNo]): (59, [ 59 59 ] )

28 (key, [pageNo, slotNo]): (60, [ 60 60 ] )

29 (key, [pageNo, slotNo]): (61, [ 61 61 ] )

30 (key, [pageNo, slotNo]): (62, [ 62 62 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 6

Left Link : 4

Right Link : -1

0 (key, [pageNo, slotNo]): (63, [ 63 63 ] )

1 (key, [pageNo, slotNo]): (64, [ 64 64 ] )

2 (key, [pageNo, slotNo]): (65, [ 65 65 ] )

3 (key, [pageNo, slotNo]): (66, [ 66 66 ] )

4 (key, [pageNo, slotNo]): (67, [ 67 67 ] )

5 (key, [pageNo, slotNo]): (68, [ 68 68 ] )

6 (key, [pageNo, slotNo]): (69, [ 69 69 ] )

7 (key, [pageNo, slotNo]): (70, [ 70 70 ] )

8 (key, [pageNo, slotNo]): (71, [ 71 71 ] )

9 (key, [pageNo, slotNo]): (72, [ 72 72 ] )

10 (key, [pageNo, slotNo]): (73, [ 73 73 ] )

11 (key, [pageNo, slotNo]): (74, [ 74 74 ] )

12 (key, [pageNo, slotNo]): (75, [ 75 75 ] )

13 (key, [pageNo, slotNo]): (76, [ 76 76 ] )

14 (key, [pageNo, slotNo]): (77, [ 77 77 ] )

15 (key, [pageNo, slotNo]): (78, [ 78 78 ] )

16 (key, [pageNo, slotNo]): (79, [ 79 79 ] )

17 (key, [pageNo, slotNo]): (80, [ 80 80 ] )

18 (key, [pageNo, slotNo]): (81, [ 81 81 ] )

19 (key, [pageNo, slotNo]): (82, [ 82 82 ] )

20 (key, [pageNo, slotNo]): (83, [ 83 83 ] )

21 (key, [pageNo, slotNo]): (84, [ 84 84 ] )

22 (key, [pageNo, slotNo]): (85, [ 85 85 ] )

23 (key, [pageNo, slotNo]): (86, [ 86 86 ] )

24 (key, [pageNo, slotNo]): (87, [ 87 87 ] )

25 (key, [pageNo, slotNo]): (88, [ 88 88 ] )

26 (key, [pageNo, slotNo]): (89, [ 89 89 ] )

27 (key, [pageNo, slotNo]): (90, [ 90 90 ] )

28 (key, [pageNo, slotNo]): (91, [ 91 91 ] )

29 (key, [pageNo, slotNo]): (92, [ 92 92 ] )

30 (key, [pageNo, slotNo]): (93, [ 93 93 ] )

31 (key, [pageNo, slotNo]): (94, [ 94 94 ] )

32 (key, [pageNo, slotNo]): (95, [ 95 95 ] )

33 (key, [pageNo, slotNo]): (96, [ 96 96 ] )

34 (key, [pageNo, slotNo]): (97, [ 97 97 ] )

35 (key, [pageNo, slotNo]): (98, [ 98 98 ] )

36 (key, [pageNo, slotNo]): (99, [ 99 99 ] )

37 (key, [pageNo, slotNo]): (100, [ 100 100 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

------------- All Leaf Pages Have Been Printed --------

-------------------------- MENU ------------------

[0] Print the B+ Tree Structure

[1] Print All Leaf Pages

[2] Choose a Page to Print

---Integer Key (for choices [3]-[5]) ---

[3] Insert a Record

[4] Delete a Record (Naive Delete)

[5] Delete some records (Naive Delete)

[6] Quit!

Hi, make your choice :3

-----------------------------

1. Insert single value

2. Insert multiple value

Make your choice(3 to exit) :

2

Please input the LOWER integer key(>=0):

101

Please input the HIGHER integer key(>=0)

2700

-------------------------- MENU ------------------

[0] Print the B+ Tree Structure

[1] Print All Leaf Pages

[2] Choose a Page to Print

---Integer Key (for choices [3]-[5]) ---

[3] Insert a Record

[4] Delete a Record (Naive Delete)

[5] Delete some records (Naive Delete)

[6] Quit!

Hi, make your choice :0

---------------The B+ Tree Structure---------------

1 90

2 5

3 3

3 4

3 6

3 7

3 8

3 9

3 10

3 11

3 12

3 13

3 14

3 15

3 16

3 17

3 18

3 19

3 20

3 21

3 22

3 23

3 24

3 25

3 26

3 27

3 28

3 29

3 30

3 31

3 32

3 33

3 34

3 35

3 36

3 37

3 38

3 39

3 40

3 41

3 42

3 43

3 44

3 45

2 89

3 46

3 47

3 48

3 49

3 50

3 51

3 52

3 53

3 54

3 55

3 56

3 57

3 58

3 59

3 60

3 61

3 62

3 63

3 64

3 65

3 66

3 67

3 68

3 69

3 70

3 71

3 72

3 73

3 74

3 75

3 76

3 77

3 78

3 79

3 80

3 81

3 82

3 83

3 84

3 85

3 86

3 87

3 88

3 91

3 92

--------------- End ---------------

-------------------------- MENU ------------------

[0] Print the B+ Tree Structure

[1] Print All Leaf Pages

[2] Choose a Page to Print

---Integer Key (for choices [3]-[5]) ---

[3] Insert a Record

[4] Delete a Record (Naive Delete)

[5] Delete some records (Naive Delete)

[6] Quit!

Hi, make your choice :4

Please input the integer key to delete:

20

-------------------------- MENU ------------------

[0] Print the B+ Tree Structure

[1] Print All Leaf Pages

[2] Choose a Page to Print

---Integer Key (for choices [3]-[5]) ---

[3] Insert a Record

[4] Delete a Record (Naive Delete)

[5] Delete some records (Naive Delete)

[6] Quit!

Hi, make your choice :5

Please input the LOWER integer key(>=0):

21

Please input the HIGHER integer key(>=0)

29

-------------------------- MENU ------------------

[0] Print the B+ Tree Structure

[1] Print All Leaf Pages

[2] Choose a Page to Print

---Integer Key (for choices [3]-[5]) ---

[3] Insert a Record

[4] Delete a Record (Naive Delete)

[5] Delete some records (Naive Delete)

[6] Quit!

Hi, make your choice :2

Please input the page number:

3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*To Print an Leaf Page \*\*\*\*\*\*\*\*

Current Page ID: 3

Left Link : -1

Right Link : 4

0 (key, [pageNo, slotNo]): (1, [ 1 1 ] )

1 (key, [pageNo, slotNo]): (2, [ 2 2 ] )

2 (key, [pageNo, slotNo]): (3, [ 3 3 ] )

3 (key, [pageNo, slotNo]): (4, [ 4 4 ] )

4 (key, [pageNo, slotNo]): (5, [ 5 5 ] )

5 (key, [pageNo, slotNo]): (6, [ 6 6 ] )

6 (key, [pageNo, slotNo]): (7, [ 7 7 ] )

7 (key, [pageNo, slotNo]): (8, [ 8 8 ] )

8 (key, [pageNo, slotNo]): (9, [ 9 9 ] )

9 (key, [pageNo, slotNo]): (10, [ 10 10 ] )

10 (key, [pageNo, slotNo]): (11, [ 11 11 ] )

11 (key, [pageNo, slotNo]): (12, [ 12 12 ] )

12 (key, [pageNo, slotNo]): (13, [ 13 13 ] )

13 (key, [pageNo, slotNo]): (14, [ 14 14 ] )

14 (key, [pageNo, slotNo]): (15, [ 15 15 ] )

15 (key, [pageNo, slotNo]): (16, [ 16 16 ] )

16 (key, [pageNo, slotNo]): (17, [ 17 17 ] )

17 (key, [pageNo, slotNo]): (18, [ 18 18 ] )

18 (key, [pageNo, slotNo]): (19, [ 19 19 ] )

19 (key, [pageNo, slotNo]): (30, [ 30 30 ] )

20 (key, [pageNo, slotNo]): (31, [ 31 31 ] )

\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*

-------------------------- MENU ------------------

[0] Print the B+ Tree Structure

[1] Print All Leaf Pages

[2] Choose a Page to Print

---Integer Key (for choices [3]-[5]) ---

[3] Insert a Record

[4] Delete a Record (Naive Delete)

[5] Delete some records (Naive Delete)

[6] Quit!

Hi, make your choice :